Discovery V5 DHT Documentation

Release 0.1.0-alpha.2

The Ethereum Foundation

Jun 03, 2021

CONTENTS

Implementation of the P2P Discoveryv5 Protocol

CHAPTER

ONE

CONTENTS

1.1 ddht package

1.1.1 Subpackages

ddht.tools package

Subpackages

ddht.tools.factories package

Submodules

ddht.tools.factories.boot_info module

ddht.tools.factories.discovery module

ddht.tools.factories.kademlia module

ddht.tools.factories.keys module

ddht.tools.factories.socket module

Module contents

Submodules

ddht.tools.v5_strategies module

Module contents

ddht.v5 package

Submodules

ddht.v5.abc module

ddht.v5.channel_services module

ddht.v5.constants module

ddht.v5.encryption module

ddht.v5.endpoint_tracker module

ddht.v5.handshake module

ddht.v5.message_dispatcher module

ddht.v5.messages module

ddht.v5.packer module

ddht.v5.packets module

ddht.v5.routing_table module

ddht.v5.routing_table_manager module

ddht.v5.tags module

ddht.v5.topic_table module

ddht.v5.typing module

Module contents

1.1.2 Submodules

1.1.3 ddht.abc module

1.1.4 ddht.app module

1.1.5 ddht.boot_info module

1.1.6 ddht.cli_commands module

1.1.7 ddht.cli_parser module

1.1.8 ddht.constants module

1.1.9 ddht.enr module

1.1.10 ddht.exceptions module

1.1.11 ddht.identity_schemes module

1.1.12 ddht.kademlia module

1.1.13 ddht.logging module

1.1.14 ddht.main module

1.1.15 ddht.typing module

1.1.16 ddht.validation module

1.1.17 ddht.xdg module

1.1.18 Module contents

1.2 JSON-RPC Endpoints

1.2.1 Common

Routing Table Stats

discv5_routingTableInfo

Fetch meta information about the routing table.

- Params: None
- Returns: `TableInfoResponse`

```
>>> rpc('discv5_routingTableInfo')
{
  'id': 0,
  'jsonrpc': '2.0',
  'result': {
    'center_node_id': '0xabcd....',
    'num_buckets': 256,
    'bucket_size': 16,
    'buckets': {...}
  }
}
```

Local ENR Management

discv5_nodeInfo

Fetch information about the local node. Delegates call to `ENRManager.enr`

- Params: None
- Returns: `NodeInfoResponse`

```
>>> rpc('discv5_nodeInfo')
{
  'id': 0,
  'jsonrpc': '2.0',
  'result': {
    'node_id': '0xabcd....',
    'enr': 'enr:-...'
  }
}
```

discv5_updateNodeInfo

Add, update, or remove a key-value pair from the local node record. To remove an existing key, update it with a value of `None`, e.g. `('0xabcd', None)`. Returns the new representation of the updated node. Delegates call to `ENRManager.update`.

- Params: `kv_pairs: List[KV_PAIRS]`
- Returns: `NodeInfoResponse`

```
>>> rpc('discv5_updateNodeInfo', [('0xabcd', '0x6789')])
{
  'id': 0,
  'jsonrpc': '2.0',
  'result': {
    'node_id': '0xabcd....',
    'enr': 'enr:-...'
  }
}
```

1.2.2 Discovery v5.1

- ANY_NODE_REPR: This can be either Node ID, enode address, or string representation of an enr.
- ENR_REPR: String representation of an enr.

Retrieve and Modify ENR Records

discv5_setENR

Write an ethereum node record to the database. Returns None if the node record has been successfully saved. Delegates call to ENRDB.set_enr.

- Params: enr: ENR_REPR
- Returns: None

```
>>> rpc('discv5_setENR', ['enr:-...'])
{
  'id': 0,
  'jsonrpc': '2.0',
  'result': None
}
```

discv5_getENR

Fetch the latest ENR associated with the given node ID. Delegates call to ENRDB.get_enr.

- Params: node_id: ANY_NODE_REPR
- Returns: GetENRResponse

```
>>> rpc('discv5_getENR', ['0xabcd...'])
{
  'id': 0,
  'jsonrpc': '2.0',
  'result': {
    'enr_repr': 'enr:-...'
  }
}
```

discv5_deleteENR

Delete a Node ID from the local database. Returns None upon successful deletion of the node record. Delegates call to ENRDB.delete_enr.

- Params: node_id: ANY_NODE_REPR
- Returns: None

```
>>> rpc('discv5_deleteENR', ['enr:-...'])
{
  'id': 0,
```

(continues on next page)

(continued from previous page)

```
  'jsonrpc': '2.0',
  'result': None
}
```

discv5_lookupENR

Fetch the ENR representation associated with the given Node ID and optional sequence number. Delegates call to `NetworkAPI.lookup_enr`.

- Params: `node_id: ANY_NODE_REPR`, `sequence_number: Optional[int]`
- Returns: `GetENRResponse`

```
>>> rpc('discv5_lookupENR', ['0xabc...', 1])
{
  'id': 0,
  'jsonrpc': '2.0',
  'result': {
    'enr_repr': 'enr:-...'
  }
}
```

ClientAPI Singular Message Sending

discv5_sendPing

Send a `PING` message to the specified node. Delegates call to `ClientAPI.send_ping`.

- Params: `node_id: ANY_NODE_REPR`
- Returns: `SendPingResponse`

```
>>> rpc('discv5_sendPing', ['enr:-...'])
{
  'id': 0,
  'jsonrpc': '2.0',
  'result': {
    'request_id': '0xabcd...',
  }
}
```

discv5_sendPong

Respond to a received `PING` message, by sending a `PONG` message to the initiating node. Delegates call to `ClientAPI.send_pong`.

- Params: `node_id: ANY_NODE_REPR`, `request_id: HexStr`
- Returns: `None`

```
>>> rpc('discv5_sendPong', ['enr:-...', '0xabcd...'])
{
  'id': 0,
  'jsonrpc': '2.0',
  'result': None
}
```

discv5_sendFindNodes

Send a `FINDNODES` request to a peer, to search within the given set of distances. Returns the request ID assigned to the request. Delegates call to `ClientAPI.send_find_nodes`.

- Params: `node_id: ANY_NODE_REPR`, `distances: List[int]`
- Returns: `HexStr`

```
>>> rpc('discv5_sendFindNodes', ['enr:-...', [1, 2, 3]])
{
  'id': 0,
  'jsonrpc': '2.0',
  'result': '0x00000000'
}
```

discv5_sendFoundNodes

Respond to a specific `FINDNODES` request with a `FOUNDNODES` response. Returns the number of batches in which the given ENRs were divided and transmitted. Delegates call to `ClientAPI.send_found_nodes`.

- Params: `node_id: ANY_NODE_REPR`, `found_nodes: List[ENR_REPR]`, `request_id: HexStr`
- Returns: `int`

```
>>> rpc('discv5_sendFoundNodes', ['enr:-...', ['enr:-...', 'enr:-...'], '0xabcd...'])
{
  'id': 0,
  'jsonrpc': '2.0',
  'result': 1
}
```

discv5_sendTalkRequest

Send a `TALKREQUEST` request with a payload to the given peer. Returns the request ID assigned to the request. Delegates call to `ClientAPI.send_talk_request`.

- Params: `node_id: ANY_NODE_REPR`, `protocol: HexStr`, `payload: HexStr`
- Returns: `HexStr`

```
>>> rpc('discv5_sendTalkRequest', ['enr:-...', '0xabcd...', '0x1234...'])
{
  'id': 0,
  'jsonrpc': '2.0',
```

(continues on next page)

(continued from previous page)

```
  'result': '0x00000000'
}
```

discv5_sendTalkResponse

Respond to a TALKREQUEST request by sending a TALKRESPONSE response. Delegates call to ClientAPI.send_talk_response.

- Params: node_id: ANY_NODE_REPR, payload: HexStr, request_id: HexStr
- Returns: None

```
>>> rpc('discv5_sendTalkResponse', ['enr:-...', '0xabcd...', '0x1234...'])
{
  'id': 0,
  'jsonrpc': '2.0',
  'result': None
}
```

NetworkAPI Round Trip Messages

discv5_ping

Send a PING message to the designated node and wait for a PONG response. Delegates call to NetworkAPI.ping.

- Params: node_id: ANY_NODE_REPR
- Returns: PongResponse

```
>>> rpc('discv5_ping', ['enr:-...'])
{
  'id': 0,
  'jsonrpc': '2.0',
  'result': {
    'enr_seq': 3,
    'packet_ip': '127.0.0.1',
    'packet_port': 30303
  }
}
```

discv5_findNodes

Send a FINDNODES request for nodes that fall within the given set of distances, to the designated peer and wait for a response. Delegates call to NetworkAPI.find_nodes.

- Params: node_id: ANY_NODE_REPR, distances: List[int]
- Returns: List[ENR_REPR]

```
>>> rpc('discv5_findNodes', ['enr:-...', [1, 2, 3]])
{
  'id': 0,
  'jsonrpc': '2.0',
  'result': ['enr:-...', 'enr:-...']
}
```

discv5_talk

Send a `TALKREQUEST` request to the designated node, and wait for its `TALKRESPONSE` response which is returned as a hexstring. Delegates call to `NetworkAPI.talk`.

- Params: `node_id: ANY_NODE_REPR`, `protocol: HexStr`, `payload: HexStr`
- Returns: `HexStr`

```
>>> rpc('discv5_talk', ['enr:-...', '0x1234...', '0x5678...'])
{
  'id': 0,
  'jsonrpc': '2.0',
  'result': '0xabcd...'
}
```

High Level *NetworkAPI*

discv5_recursiveFindNodes

Lookup a target node within in the network. Delegates call to `NetworkAPI.recursive_find_nodes`.

- Params: `node_id: ANY_NODE_REPR`
- Returns: `List[ENR_REPR]`

```
>>> rpc('discv5_recursiveFindNodes', ['enr:-...'])
{
  'id': 0,
  'jsonrpc': '2.0',
  'result': ['enr:-...', 'enr:-...']
}
```

discv5_bond

Bond with the given node to ensure liveness. Delegates call to `NetworkAPI.bond`.

- Params: `node_id: ANY_NODE_REPR`
- Returns: `bool`

```
>>> rpc('discv5_bond', ['enr:-...'])
{
  'id': 0,
  'jsonrpc': '2.0',
```

(continues on next page)

(continued from previous page)

```
  'result': True
}
```

1.3 Release Notes

1.3.1 v0.1.0-alpha.1

- Launched repository, claimed names for pip, RTD, github, etc

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex